



**GH**

高等学校工科电子类教材

# 计算机软件实践教程

## 基础软件部分

(第二版)

郭浩志 主编

西安电子科技大学出版社

高等学校教材

# 计算机软件实践教程

(基础软件部分)

(第二版)

郭浩志 主编

西安电子科技大学出版社

1 9 9 3

(陕)新登字 010 号

## 内 容 简 介

本书系《计算机软件实践教程》和《应用软件实践教程》的修订版本,旨在介绍我国高等院校计算机专业各有关课程上机实习的内容、方法和要求。各章分别给出适当示例和实习题,但不局限于具体的计算机和程序语言。全书比较全面地介绍了当前我国高等院校主要软件课程的实践性教学环节。全书共分三册。上册为基础软件部分,中册为系统软件部分,下册为应用软件部分。本册系上册,除了介绍基本知识和实习入门训练外,还分别给出 PASCAL、C、PC 汇编、VAX 汇编、离散数学、数据结构、数值分析和程序设计方法学等课程的实践性教学环节。本书的组织既注意到内容的系统性和形式的多样性,又力使各章内容具有相对的独立性,使每一章可以单独用来指导该门课程的实习。

本书系高等院校计算机专业的教学用书,供大学生在整个大学期间使用,也可作为教师、研究生或工程技术人员的参考书。

高等学校教材  
计算机软件实践教程  
基础软件部分  
(第二版)  
郭浩志 主编  
责任编辑 汪海洋

---

西安电子科技大学出版社出版

陕西省大荔县印刷厂印刷

陕西省新华书店发行 各地新华书店经售

开本 787×1092 1/16 印张 26 14/16 字数 638 千字

1985 年 6 月第 1 版 1993 年 12 月第 2 版 1993 年 12 月第 1 次印刷 印数 1—5 000

---

ISBN 7-5606-0239-8/TP·0093 (课)

定价: 12.30 元

## 出版说明

根据国务院关于高等学校教材工作的规定，我部承担了全国高等学校和中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从1978~1990，已编审、出版了三个轮次教材，及时供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要，贯彻国家教委《高等教育“八五”期间教材建设规划纲要》的精神，“以全面提高教材质量水平为中心，保证重点教材，保持教材相对稳定，适当扩大教材品种，逐步完善教材配套”，作为“八五”期间工科电子类专业教材建设工作的指导思想，组织我部所属的九个高等学校教材编审委员会和四个中等专业学校专业教学指导委员会，在总结前三轮教材工作的基础上，根据教育形势的发展和教学改革的需要，制订了1991~1995年的“八五”（第四轮）教材编审出版规划。列入规划的，以主要专业主干课程教材及其辅助教材为主的教材约300多种。这批教材的评选推荐和编审工作，由各编委会或教学指导委员会组织进行。

这批教材的书稿，其一是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会（小组）评选择优产生出来的，其二是在认真遴选主编人的条件下进行约编的，其三是经过质量调查在前几轮组织编定出版的教材中修编的。广大编审者、各编审委员会（小组）、教学指导委员会和有关出版社，为保证教材的出版和提高教材的质量，作出了不懈的努力。

限于水平和经验，这批教材的编审、出版工作还可能有缺点和不足之外，希望使用教材的单位，广大教师和同学积极提出批评和建议，共同为不断提高工科电子类专业教材的质量而努力。

机械电子工业部电子类专业教材办公室

## 再 版 前 言

本书是高等院校工科电子类计算机专业统编教材，由中国电子工业总公司计算机教材编委会审定，并推荐出版。责任编委为浙江大学俞瑞钊教授。

全书按计算机编委会审定的编写大纲进行编写和审阅。由西安电子科技大学蔡希尧教授主审，国防科技大学郭浩志教授主编。

《计算机软件实践教程》和《应用软件实践教程》两种全国统编教材自 85、86 年出版以来，两次重印，得到国内高校的欢迎和专家教授的充分肯定，各界一致认为：“该书是国内公开发行的第一本软件实践教材，为国内大多数工科院校计算机课程的教学和软件人材的培养，发挥了重要的作用。属国内首创。”修订版仍以计算机教学计划中对大学本科生培养目标的基本要求为出发点，着眼于加强对学生的基本技能和科学作风的培养和训练。全书着重介绍诸软件课程上机实习的目的、要求、内容和方法。诸实习内容至少都有一个示例，并提供若干实习题供各校选择，有些题目还给出必要的提示和思考题。每章最后均列出该章的主要参考文献。

为适应计算机科学技术和高等教育事业的蓬勃发展，此次再版，在保留原书特色的基础上，内容做了很大更新，主要变动如下：

- 增加离散数学、软件工程、人机界面、人工智能、专家系统和算法设计与分析等内容；
- 将 PDP 汇编改为 PC 汇编和 VAX 汇编；
- 将高级语言程序设计改为 PASCAL 和 C 两章；
- 删去系统程序设计；
- 对操作系统和编译分别增加大作业。

即使题目不变的各章，其内容大多数也做了很大的更新。

修订版是在总结我国若干所理工科大学软件课程近几年教学实习的实践经验和比较广泛搜集其它院校有关资料和意见的基础上编写的。编写时尽量考虑对全国大多数院校的通用性。各校在采用本书时，宜根据各自的教学大纲和实际条件，结合具体的计算机和程序语言，做适当的选择和补充。

本书拟分三册出版。上册为基础软件部分，中册为系统软件部分，下册为应用软件部分。本册为上册，共分十一章，依次为概述、基本知识、入门训练、PASCAL、C、PC 汇编、VAX 汇编、离散数学、数据结构、数值分析和程序设计方法学等。

参加上册编写的有郭浩志（第一至第四章），孙志挥、宋明龙（第五章），王保恒（第六章），陈怀义（第七章），张强（第八章），朱静华（第九章），黎建兴（第十章），董逸生（第十一章）。

在本书编写过程中，得到了西安电子科技大学、浙江大学、东南大学、东北大学、西北工业大学、上海交通大学、清华大学、国防科技大学许多领导、教师和学生的关心和支持，在此表示谢意。

本书虽是再版，仍感不很适应科技与教育的飞速发展。加上修订时间仓促，编者水平不齐，错误在所难免，热诚欢迎读者批评指正。

编 者

# 目 录

|                                        |     |
|----------------------------------------|-----|
| <b>第一章 概述</b> .....                    | 1   |
| 第一节 软件实习的原则、目的与要求.....                 | 1   |
| 第二节 软件实习的组织与方式.....                    | 2   |
| <b>第二章 基本知识</b> .....                  | 4   |
| 第一节 软件实习环境.....                        | 4   |
| 第二节 算法及其表示方法.....                      | 6   |
| 第三节 程序设计方法.....                        | 8   |
| 第四节 程序设计质量 .....                       | 14  |
| 第五节 软件工程化 .....                        | 21  |
| <b>第三章 入门训练</b> .....                  | 27  |
| 第一节 指法练习 .....                         | 27  |
| 第二节 联机终端练习 .....                       | 29  |
| <b>第四章 PASCAL 语言程序设计实习</b> .....       | 35  |
| 第一节 概述 .....                           | 35  |
| 第二节 语言环境初识与简易程序设计 .....                | 36  |
| 第三节 循环程序设计 .....                       | 39  |
| 第四节 子程序设计 .....                        | 42  |
| 第五节 数组与集合的程序设计 .....                   | 45  |
| 第六节 递归程序设计 .....                       | 50  |
| 第七节 记录程序设计 .....                       | 55  |
| 第八节 文件程序设计 .....                       | 60  |
| 第九节 指针程序设计 .....                       | 64  |
| 第十节 综合实习 .....                         | 68  |
| <b>第五章 C 语言程序设计实习</b> .....            | 87  |
| 第一节 概述 .....                           | 87  |
| 第二节 基本程序控制语句与流程控制 .....                | 88  |
| 第三节 指针 .....                           | 96  |
| 第四节 复杂数据类型——数组、结构和联合.....              | 101 |
| 第五节 位操作和特殊表达式.....                     | 108 |
| 第六节 函数与程序结构.....                       | 113 |
| 第七节 文件与综合程序设计.....                     | 119 |
| <b>第六章 IBM PC 系列微机汇编语言程序设计实习</b> ..... | 132 |
| 第一节 概述.....                            | 132 |
| 第二节 直接程序设计.....                        | 132 |
| 第三节 分枝程序设计.....                        | 134 |

|     |                        |     |
|-----|------------------------|-----|
| 第四节 | 循环程序设计.....            | 138 |
| 第五节 | 子程序程序设计.....           | 144 |
| 第六节 | 宏指令与条件汇编.....          | 149 |
| 第七节 | 调用 DOS、BIOS 系统功能 ..... | 156 |
| 第八节 | 中断服务程序.....            | 162 |
| 第七章 | VAX 汇编语言程序设计实习 .....   | 174 |
| 第一节 | 概述.....                | 174 |
| 第二节 | 顺序程序设计.....            | 177 |
| 第三节 | 分枝程序设计.....            | 180 |
| 第四节 | 输入输出程序设计.....          | 184 |
| 第五节 | 循环程序设计.....            | 189 |
| 第六节 | 子程序设计.....             | 199 |
| 第七节 | 宏指令程序设计.....           | 210 |
| 第八节 | 综合练习.....              | 215 |
| 第八章 | 离散数学实习.....            | 230 |
| 第一节 | 概述.....                | 230 |
| 第二节 | 集合.....                | 231 |
| 第三节 | 关系和函数.....             | 233 |
| 第四节 | 组合论.....               | 236 |
| 第五节 | 图论.....                | 238 |
| 第六节 | 代数结构.....              | 242 |
| 第九章 | 数据结构实习.....            | 246 |
| 第一节 | 概述.....                | 246 |
| 第二节 | 线性数据结构及其顺序表示.....      | 246 |
| 第三节 | 链表.....                | 260 |
| 第四节 | 串.....                 | 283 |
| 第五节 | 树.....                 | 298 |
| 第六节 | 图.....                 | 311 |
| 第七节 | 排序和检索.....             | 327 |
| 第八节 | 文件.....                | 342 |
| 第十章 | 数值分析实习.....            | 357 |
| 第一节 | 概述.....                | 357 |
| 第二节 | 插值.....                | 358 |
| 第三节 | 数值积分.....              | 360 |
| 第四节 | 常微分方程数值解.....          | 367 |
| 第五节 | 非线性方程求根.....           | 373 |
| 第六节 | 数值逼近.....              | 376 |
| 第七节 | 线性代数方程组的数值解.....       | 379 |
| 第八节 | 矩阵特征值和特征向量的计算.....     | 386 |

|                             |     |
|-----------------------------|-----|
| <b>第十一章 程序设计方法学实习</b> ..... | 392 |
| 第一节 概述.....                 | 392 |
| 第二节 结构化程序设计.....            | 392 |
| 第三节 程序变换.....               | 407 |

# 第一章 概 述

## 第一节 软件实习的原则、目的与要求

### 一、原则

1. 软件实习是计算机科学与工程有关专业的必修内容。有关教学大纲要明确指出软件实习的地位、目的与要求，使课堂教学、课外练习与上机实习，以及各有关课程之间的实习有机地结合起来。

2. 明确各课程软件实习的分工、难点、重点和衔接关系，尽量减少不必要的重复。做到目的明确、要求合理、联系紧密、循序渐进。软件实习，要始终重视程序设计基本方法、技巧和风格的训练，逐步提高要求和学生的素质。

3. 确保学生在校期间有较充分的上机实习时间，每个学期都有所安排，使上机实习不断线。

4. 精心设计和安排好每一门课程的每一次作业的实习。

5. 实习内容一方面要有基本要求，另一方面要尽量考虑留有选择余地，使不同程度的学生均能适用。

### 二、实习目的与要求

通过软件课程的学习和一系列的实习，使学生能熟练地运用汇编语言和至少一种高级语言独立完成有关软件（系统软件、应用软件）的设计、编制、上机调试和运行，加深对课堂讲授的概念和原理的理解，加强程序设计基本方法的训练，初步具备利用计算机进行科学研究的能力。这些能力的培养是专业培养目标的一个重要组成部分。

在整个学习期间，力争每学期都结合有关课程安排学生上机实习。在制订教学计划时，应将各有关实习内容，比较均匀而有机地安排在各个学期。在不同课程中，安排不同要求的适量的上机学习。从当前情况来看，总实习时间（不包含生产实习与毕业设计的实习）以不低于 200 h 为宜。

根据需要可设置一至三种高级语言和一两种汇编语言的程序设计，作为有关软件实习课的先行课程。各种程序设计课程的实习要求，应由简到繁，由易到难，逐步加大实习量，不断加深实习深度。各种语言程序设计课程的实习是其它软件实习课程的实习基础，宜单独实施，而且，一方面要确保有较充分的实习时间，另一方面要加强实习指导，严格要求。

## 第二节 软件实习的组织与方式

### 一、实习教学环节

软件实习宜根据各有关课程的特点,在不同教学环节中分别采用各种不同的方式,以培养和不断提高学生的软件基本技能。这些实习教学环节主要指基础训练、课程应用、课程设计、生产实习和毕业设计。

#### 1. 基础训练

各种语言程序设计课程的实习是整个软件实习的基础,应扎扎实实安排好各种语言的程序结构、重要语句(指令)、数据类型、机制和特色以及程序设计基本方法的初步训练。尤其最初一、两次入门性训练,更应精心安排,实习量不宜过大,让学生先从模拟入手。但严格要求,加强指导,使学生在受到比较正规而扎实的基础训练的过程中,逐步达到独立上机的目的。

#### 2. 课程应用

以各种程序设计语言为工具,结合《数值分析》和《数据结构》等课程,进行数学和数据结构等领域若干基本算法的实习,使学生初步掌握程序设计语言在数值计算和非数值计算方面的若干应用。要求学生独立编制和调试出有关应用问题的程序,并力求程序质量较高,资料齐全。

#### 3. 课程设计

在诸如《编译原理》、《操作系统》、《数据库》、《软件工程》和《软件环境与工具》这些课程学习告一段落或结束时,可集中一段时间,以小组为单位,集体完成中等规模的大作业——简易的系统软件。例如简单的编译程序、操作系统、数据库,以培养学生设计、开发软件的能力,以及集体完成中等规模软件所应具备的组织能力和协作能力。

具体实施时,可从上列课程中选择其中的一、两种安排课程设计,其余仍可按“课程应用”的教学形式进行实习。

#### 4. 生产实习

该教学环节在专门的教学实习时间内进行,仍以3.中所列那些课程为实习内容,但强调不能停留在单纯为了验证某些原理的常规实习上,宜从科研、生产等国民经济实际环节中选题,让学生集体合作设计、编制和调试出完整可用的软件。

#### 5. 毕业设计

此教学环节中的实习,可与生产实习或科技开发活动通盘考虑。

### 二、实习、练习与课堂讲授

课堂讲授为实习和练习指明了范围、内容和方法。而实习和练习则将进一步巩固和加深课堂讲授的知识,它们均是教学大纲中的重要组成部分。

平时练习题多而宽,各部分内容都可照顾到,但缺点是“纸上谈兵”,学生无法上机验证程序是否正确。软件实习是平时练习的继续和深入,学生亲自调试出一个程序往往胜过在纸上编写十个程序。不能以静态的程序练习替代动态的程序实习。

在加强实践环节的情况下,弄不好容易导致学时膨胀和学生负担过重的现象出现。应采取相应的措施。除了努力提高课堂教学质量,避免填鸭式教学和在课外变相增加学时外,还可有意识地把实习题与练习题结合起来,以减轻实习的准备量。此外,课堂讲授时间与实习、练习时间,宜按一定比例安排。汇编语言与高级语言程序设计的授课时间与实习练习时间的比例约为 3:1~2:1,而其余课程可为 6:1~4:1。

### 三、教师指导

在各种语言程序设计的基础训练阶段,教师宜加强上机指导,尤其是指导学生充分利用现代计算机系统所提供的各种手段,例如交互式联机调试软件,诊断和改正程序中的错误,并学会验证程序正确性的若干技术,使学生受到比较系统和严格的训练。此后,可有意识地逐步减少辅导,努力培养学生的独立工作能力。在课程设计和生产实习阶段,实习可以小组为单位,使学生受到集体完成中等规模软件所应具备的组织能力和协作能力的训练。此外,教师应允许优秀学生或有特长的学生自由选题,并指导其完成带探索性和创造性的独立研究活动。

### 四、实习程序规模

实习题应逐步由简到繁,由易到难,由小到大;实习量应逐步由少到多。各种语言程序设计的实习,初次可选择十几行、几十行指令(语句)规模的程序,甚至允许学生依样画葫芦,逐步加到几百行。到了课程设计和生产实习阶段,实习量宜不低于 500 行。

实习程序的类型,从练习性程序过渡到各种应用程序和系统程序,从无接口关系的程序到有软件接口(包括窗口、菜单)的系统实用程序,再到有软硬件接口的系统程序。从单纯为了验证某些原理的假想程序到实际问题的程序。从一般的程序编制到软件的设计、开发和维护。

### 五、多种形式的考核制度

1. 每次实习均须要求每人(组)完成上机实习报告。实习报告的主要内容,除了源程序和流程图外,还应要求学生书写有关设计、调试和使用等方面的情况。最好每次实习均印发统一格式的实习报告单给学生填写,以便存档。

2. 抽样检查。教师从磁盘调出学生编写的程序,输入数据,检查结果的正确性。并视情况适当设置故障和问题,以激发学生智力,促其深入考虑问题。

3. 现场考核。汇编语言和第一高级语言的实习,可逐步试行现场考核制度,即要求学生终端上限时限量按质完成试题。

4. 在笔试中增加若干与实习内容有关的试题。

5. 实习应评定成绩。譬如,根据不同课程和教学要求,实习成绩可分别占该门课程总成绩的 10%~30%。

### 主要参考文献

- [1] 郭浩志主编:《计算机软件实践教程》,西北电讯工程学院出版社,1985.6。
- [2] 郭浩志主编:《应用软件实践教程》,西北电讯工程学院出版社,1986.6。

## 第二章 基本知识

### 第一节 软件实习环境

软件实习,与一般利用计算机算题一样,是在整个计算机系统的控制与支持下进行的。计算机系统指的是计算机硬件(简称硬件)和计算机软件(简称软件)。

#### 一、计算机硬件

计算机的作用,主要是按人的意图代替人的部分脑力劳动。人处理问题,首先通过眼、耳、鼻等感觉器官接受外界信息,记入大脑,再经过分析、思考和计算,得出结论,最后用各种方式,如语言、文字书写等,将结果告诉外界。类似地,计算机首先从外界接受原始信息(如源程序、输入数据),经过计算、判断,再将结果送出到外界。这个过程是在硬件与软件密切配合下完成的。对于大脑的三种功能(指记忆、分析、思考和计算,按一定规则对信息进行控制),计算机相应存有存储器、运算器和控制器。这三部分统称为中央处理器,简称主机或 CPU。

运算器负责对各种不同结构的代码进行各种算术运算和其它非数值运算。主存储器(内存)是一种记忆装置,直接与运算器发生联系。它能暂存用户程序、原始数据、中间结果、最终结果以及计算机系统软件程序。控制器主要指中央控制器和指令控制器,前者指挥全机各部件相互协调地工作,后者控制一条指令的执行过程。

主存的容量往往不大够用,通常还配置外存储器(外存),又称辅助存储器。常用的有磁盘和磁带。外存不直接与运算器发生联系,只与内存打交道,充当“后台”的角色。计算机系统把内存中暂时不用的代码记入外存,让内存空出来存放当时急用的代码。外存的代码,在需要时再一批批地调入内存。

把信息从外部介质输入到内存或从内存把信息输出到外部介质的过程称为输入输出。它是通过输入输出通道(简称通道)来完成的。通道是中央处理部件与输入输出设备之间的桥梁,它起到一台小处理机的作用。常用的输入设备有键盘输入器、纸带输入机等等。常用的输出设备有行打印机、显示装置、绘图仪等等。带显示装置的键盘设备作为计算机的终端设备,主要用于人机通讯。大多数输入输出设备(统称为外部设备)在程序控制下进行操作,即由程序的指令选取所需设备,规定是输入还是输出,并指明信息传输时内存起始地址和传输数量。少数外部设备是手动的,如键盘、光笔等,将信息直接送至内存。

#### 二、计算机软件

早期计算机是台裸机,完全是由物理设备组成,仅有机器语言,即代码指令。用机器语言编程上机,人工干预频繁,时间耗费多,不能充分发挥计算机效率,使用范围不广泛。后来,人们利用计算机能高速完成复杂逻辑运算这一特点,事先编制好一些程序,随计算机一同交付用户使用。之后逐步发展成为计算机系统的必备部分。任一用户题目的执行,都

是在计算机物理装置（硬件）和上述计算机程序（软件）协同控制和支持下完成的。软件代替了原先必须由人承担的部分工作，实现了合理组织整个解题过程，简化或代替解题工作流程的有关环节，从而达到既充分发挥机器效率又大大方便用户这两个目的。

### 1. 操作系统

任一程序的运行，既要依赖各种硬件资源（指处理机、主存、外部设备等），又要用到各种软件资源（指各种实用程序、应用程序、数据库等等）。操作系统是担负组织和管理各种软硬件资源，使之协调一致地高效地完成各种各样任务的一种程序。按照传统的资源管理的观点，操作系统可由 4 大部分组成：

(1) 处理机管理程序 其主要功能是确定哪个用户的程序将在 CPU 上运行。处理机分两级管理。第一级为作业（用户题目）调度。在多终端分时系统中，CPU 轮流执行各终端用户的程序。在实时系统中，CPU 先执行有实时要求的程序，空闲时再执行其它程序。在批量多道系统中，可根据各种因素选取程序进入运行，以便既能充分满足用户的要求，又能最大限度地提高计算机的利用率。当被选中的若干个作业进入运行后，操作系统再进一步进行第二级调度（进程调度）。进程处于运行、就绪、封锁 3 种状态之一。操作系统负责管理和转换这些状态。

(2) 存贮器管理程序 其主要功能是负责将主存贮器分配给进程使用；在多道系统中对主存实行保护；还可提供虚拟存贮，使用户在使用计算机时不必考虑主存容量的限制。

(3) 设备管理程序 其主要功能是将外部设备分配给进程使用，并对外设实现各种具体的操作。此外，还可实行假脱机的控制。

(4) 信息管理程序 其主要功能是可靠地保存贮存在外存中的大量信息（通常以文件、卷宗的形式存放），并按用户的要求对文件进行各种操作（如建立、撤销、打开、关闭、读、写等等）。

操作系统本身是计算机硬件和用户之间的接口。操作系统与用户之间的接口主要包括命令接口和编程接口（如系统调用接口，程序库接口）。自 70 年代末以来，操作系统的用户接口变得越来越重要和引人注目，实际上已成为操作系统不可缺少的内容。如在 80 年代得到广泛流行的 UNIX 操作系统中的 shell 语言，既是命令语言又是程序设计语言，其基本成分是操作系统的命令或程序。

### 2. 系统实用程序

这是一套协助管理、维护和使用计算机的程序。在终端分时系统中，它们一般都各自配有一组命令，以利于用户交互使用。下面简单介绍一些常见的系统实用程序。

(1) 汇编程序 它是一种翻译程序，专门负责把用汇编语言书写的源程序翻译成目标程序（模块）。汇编语言是一种紧紧依赖于具体计算机的低级语言。其核心部分的符号机器指令与计算机原有的代码机器指令一一对应。汇编程序一般通过两遍扫描（第一遍进行符号定义，第二遍生成目标代码）完成翻译。

(2) 编译程序 它也是一种翻译程序，专门负责把用高级语言（如 FORTRAN, C, PASCAL 等）书写的源程序翻译成目标程序（模块）。高级语言是一种独立于具体计算机的语言。编译程序是一种比较复杂的加工程序。一般经过词法分析、语法分析、语义分析、优化和目标代码生成等 5 个阶段的处理，最后译成能尽量发挥硬件资源效率，并且可在计算机上执行的目标程序。

(3) 编辑程序 它提供对任一字符串进行定位、增添、替换、删去、显示、打印等的功能,形成或修改源程序文件、数据文件、资料表格和其它信息。

(4) 连接程序 它将汇编程序或编译程序或二者共同产生的一个或多个目标模块与其它介质(如各种外存和输入设备)上的一个或多个目标模块连接成一个可在计算机上执行的程序。

(5) 调试程序(查错程序) 它允许用户随时设置断点,动态地测试、修改和输出有关主存单元的内容,以便在目标程序一级进行查错和改错。

此外,还有很多其它系统实用程序,如文件转换程序,文件转贮程序,文件比较程序,分类排序程序,目标模块修补程序,库管理程序等等,在此不一一赘述。

### 3. 应用程序

为科学技术、工业、农业、商业和经济等各个领域而编制的有关计算、设计、控制、管理、实验以及各种事务处理的程序,统属于应用程序。

## 第二节 算法及其表示方法

### 一、什么是算法?

我们用一个例子来说明。

例 试计算两个正整数  $m$  和  $n$  的最大公因子。

求两个给定正整数  $m$  和  $n$  (设  $m > n$ ) 的最大公因子,可通过反复执行下列 3 个操作来实现:求  $m$  和  $n$  的余数  $r$ ;变换  $m$  和  $n$  值(以  $n$  取代  $m$ ,以  $r$  取代  $n$ );判别  $r$  是否为零。在此过程中,一旦发现  $r=0$ , $m$  即为最大公因子;否则,重复执行以上 3 个操作。我们可将计算两个正整数  $m$  和  $n$  的最大公因子的整个过程分为以下 5 步:

第一步:读入两个正整数  $m$  和  $n$  (设  $m > n$ )

第二步:求  $m$  和  $n$  的余数  $r$

用  $\text{mod}(m, n)$  表示求余数,即

$$r = \text{mod}(m, n) = m - \lfloor m/n \rfloor * n$$

这里用  $\lfloor m/n \rfloor$  表示小于或等于  $m/n$  的最大整数。例如:  $m=24$ ,  $n=9$ , 则

$$\begin{aligned} r &= \text{mod}(24, 9) = 24 - \lfloor 24/9 \rfloor * 9 \\ &= 24 - 2 * 9 = 6 \end{aligned}$$

第三步:变换  $m$  和  $n$  值

$$m := n; \quad n := r$$

第四步:判别  $r$  是否为零

如果  $r=0$ ,则  $m$  为最大公因子,转入第五步;如果  $r \neq 0$ ,则返至第二步,重复第二、第三和第四步操作。

第五步:输出  $m$  值,即为最大公因子。停止。

从第一步开始,一步一步执行,直至  $r=0$ ,转至第五步,输出最大公因子,然后停止。例如,  $m=24$ ,  $n=9$ , 执行过程如表 2.2.1 所示。

表 2.2.1

| 步 骤 | 各变量的变化 |     |     | 步 骤 | 各变量的变化 |     |     |
|-----|--------|-----|-----|-----|--------|-----|-----|
|     | $m$    | $n$ | $r$ |     | $m$    | $n$ | $r$ |
| 第一步 | 24     | 9   | 未定义 | 第四步 | 6      | 3   | 3   |
| 第二步 | 24     | 9   | 6   | 第二步 | 6      | 3   | 0   |
| 第三步 | 9      | 6   | 6   | 第三步 | 3      | 0   | 0   |
| 第四步 | 9      | 6   | 6   | 第四步 | 3      | 0   | 0   |
| 第二步 | 9      | 6   | 3   | 第五步 | 3      | 0   | 0   |
| 第三步 | 6      | 3   | 3   |     |        |     |     |

如果给定另一组  $m$  和  $n$  值, 遵循上述过程, 又可得到另一确定的结果。上面叙述的求两个正整数  $m$  和  $n$  的最大公因子的过程具体给出了一个算法, 它就是所谓的辗转相除法。

## 二、算法的表示法

算法只对解问题的方法进行了描述, 但是无法将它直接输入计算机, 以完成相应的计算 (无论是对数值问题还是非数值问题)。程序设计就是用程序去表示和实现算法。随后, 将程序输入计算机, 由计算机按照程序进行计算, 以求得问题的解。因此, 在编写程序之前, 必须清晰地描述算法。算法表示的方法通常有两类, 一类是流程图 (框图), 另一类是非形式的算法描述语言。

### 1. 流程图

描述算法和信息处理流程的图形称为流程图。它是由各种基本图形组成的。

仍以求两个正整数  $m$  和  $n$  的最大公因子的辗转相除法为例, 其流程图如图 2.2.1 所示。

图中椭圆形图形表示流程图的开始或结束。平行四边形表示输入或输出。矩形表示一般处理 (如计算)。菱形表示判别或检查。箭头表示流程图的路径和方向。流程图的基本图形有国家标准, 详见本章第五节图 2.5.1。

### 2. 非形式算法描述语言

它酷似描述程序的高级语言。为便于阅读程序, 只将算法的核心部分采用高级语言的语句描述, 其余部分或用自然语言 (如汉语、英语等) 描述, 或省略 (如说明语句)。对于上面求两个正整数最大公因子的例子, 可用非形式算法描述语言描述如下:

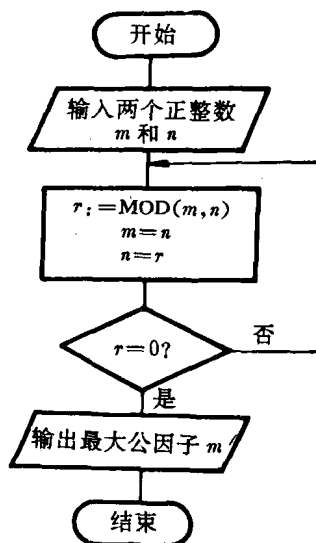


图 2.2.1 求最大公因子流程图

```

{Finding the greatest common divisor given two positive integers m and n}
BEGIN
  输入 m 和 n;
  REPEAT
    r:=MOD (m, n); {r is a remainder.}
    m:=n;
    n:=r
  UNTIL r=0;
  输出 m
END

```

可以看出,用非形式算法描述语言所描述的程序,算法核心清晰明确,便于阅读理解,但程序并不细致,若将它进一步加细为用某种高级语言表示的程序,是不难的。比如,用 PASCAL 语言描述上列算法的程序,请见本章第四节。

### 第三节 程序设计方法

#### 一、逐步求精法

从问题的体系结构、解决策略和要求出发,先提出一个解决办法,然后对此办法逐步补充加细,使之不断明确和完善,直至整个问题可用某种程序设计语言明确地、完全地描述出来为止。人们一开始往往比较习惯于用公式,甚至用某种自然语言(如汉语)来描述问题,但这比较粗糙。对解决办法逐步精细化的技术有 3 种:第一种叫分割技术,它将一个问题划分为若干互相独立的小问题,然后依次解决每个小问题;第二种叫递推技术,每使用此方法一次,就能使问题朝着彻底解决的方向推进一步,反复使用此方法,就能得到最终的解;第三种叫分析技术,它将问题进行严格分类,对不同问题采用不同的解法。这三种技术的使用也会为以后的程序编制打下良好的基础。实际上,第一种技术针对复合语句,第二种技术针对各种循环语句,第三种技术针对各种条件语句。可以说,任一问题的解决都是综合运用这 3 种技术的结果。

逐步求精法的特点是,考虑问题时先整体后局部,先抽象后具体。因此,这种方法是一种自顶向下法。此方法安全可靠,最终得到的程序的总体结构完整清晰,含义明确,易读性、易理解性均佳,也利于程序正确性的证明。

#### 例 求素数

为确定  $[1, n]$  ( $n$  为非 1 的正整数)范围内的所有素数,可以按递增次序查看自然数集合,从而选出满足素数定义的所有素数。该题目可用一个算法描述语言的程序表述如下:

```

PROCEDURE fprime
  {Finding primes 2, 3, 5, ...}
BEGIN
  read (n);

```

```

number := 1; {number is natural numbers 1, 2, 3, 4, ...}
WHILE number < n DO
  BEGIN
    number 是下一个素数;
    write (number)
  END
END {end fprime}

```

“number 是下一个素数”是一个汉语句子。但它尚不具体，需要进一步精细化。为了顺序生成下一个素数，首先要按递增顺序生成自然数，然后按素数定义判定该自然数是否为素数。重复上述操作，直至选出下一个素数为止。为此，引入逻辑变量 prim。若 prim 为真，表示 number 是下一个素数，否则查看下一个自然数。为进一步展开“number 是下一个素数”，可反复执行下列语句序列：

```

{Finding the next prime}
number := number + 1;
prim := number 是一个素数.
{If the natural number is a prime,
  then prim := TRUE. }

```

其中“number 是一个素数”表示按素数定义选出素数。number 是素数，当且仅当 number 只能被 1 及其自身整除时，即若被 2、3、4、…、number-1 除时，其余数均不为零。令 k 可取值 2、3、4、…、number-1，lim 为 number-1。这样，“prim := number 是一个素数”还可加细。为进一步展开“number 是下一个素数”，可反复执行下列语句序列：

```

{Finding the next prime}
number := number + 1;
k := 2; {k —— 2, 3, 4, ..., number-1.}
lim := number - 1; {lim —— a limit value of k.}
REPEAT {Is the number a prime?}
  IF number 能被 k 整除
    THEN prim := FALSE
  ELSE BEGIN
    k := k + 1;
    prim := TRUE
  END
UNTIL NOT(prim) OR (k 达到 lim)

```

上列“number 能被 k 整除”可用下列逻辑表达式进一步精细为：

$$\text{MOD}(\text{number}, k) = 0$$

若余数为零，则上列逻辑表达式成立。此时，number 不是素数，prim 为假，因而需顺序查看下一个自然数。若余数不为零，上列逻辑表达式不成立，number 可能是素数，此时 k 应加